

CMPT125, Fall 2021

Lab exam - Version 3

Wednesday, November 17, 2021, 3:30pm-4:20pm

You need to implement the functions in ***labexam.c***.

Submit only the **.c** file to Canvas

Canvas Assignment - Lab Exam 15:30-16:20.

You have 50 minutes to solve all 3 problems.

The maximal score is 20 points.

The exam will be graded both **automatically** and by **reading your code**.

You can run your code using

```
>> make
```

```
>> ./run_test_v3
```

Correctness: Make sure that your code compiles without warnings/errors, and works as expected.

Readability: Your code should be readable. Add comments wherever necessary. If needed, write helper functions to break the code into small, readable chunks.

Compilation: Your code **MUST** compile in CSIL with the Makefile provided. If the code does not compile in CSIL, the grade on the assignment is 0 (zero). Even if you can't solve a problem, make sure it compiles.

Helper functions: If necessary, you may add helper functions to the .c file.

main() function: do not add main(). Adding main() will cause compilation errors, as the main() function is already in the test file.

Using printf()/scanf(): Your function should have no unnecessary printf() statements. They may interfere with the automatic graders.

Warnings: Warnings during compilation will reduce points. More importantly, they indicate that something is probably wrong with the code.

Testing: An example of a test file is included.

Your code will be tested using the provided tests as well as additional tests.

You are *strongly encouraged to write more tests* to check your solution is correct, but you don't need to submit them.

Question 1 [7 points]

Write a function that gets an array of strings, and a char c, and counts the total number appearances of c in all strings. For example:

- `count_c(["Hello", "hello", "yes", "No"], 'o')` should return 3
- `count_c(["ABCD", "ABBC", "BBAB", ""], 'B')` should return 6

```
// gets an array of strings of length n, and a char c
// counts the total number of appearances of c in all strings
int count_c(const char** ar, int n, char c);
```

Question 2 [6 points]

Write a function that gets two arrays of ints and counts how many elements appear in both arrays. You may assume all values in each array are different. For example:

- `count_common([1, 2, -3], [4, -3, 1])` needs to return 2
- `count_common([1, 6, 8], [0, 5, 4, 9, -3])` needs to return 0
- `count_common([3, 2, 1], [4, 1, 2, 3])` needs to return 3

```
// gets ar1 of length n1, and ar2 of length n2
// returns the number of common elements in the two arrays
int count_common(const int* ar1, int n, const int* ar2, int n2);
```

Question 3 [7 points]

Write a function that gets a linked list of ints and returns the largest difference (in absolute value) between two consecutive nodes. For example

- On input 1 -> 3 -> 8 -> 4 -> 3 it returns 5 because $|8-3| = 5$ is the largest gap
- On input 11 -> 13 -> 14 -> 13 it returns 2 because $|11-13| = 2$ is the largest gap
- On input 20 -> 21 -> 22 -> 23 -> 24 it returns 1 because all gaps are 1

* If list is empty or has one element, the function needs to return -1

* You may not modify the list.

```
// gets a linked list of ints
// returns the largest gap between two consecutive nodes
int largest_diff(const LL_t* lst);
```